Introduction

Le programme ANALOG reprend les idées présentées par I. XENAKIS dans son livre "Musiques formelles" p 56 à 131 et plus particulièrement ce qui concerne la réalisation de ANALOBIQUE A et ANALOBIQUE B.

Le programme a été écrit et réalisé entre juin 73 et décembre 74 dans le cadre du CEMAMu et avec l'aide de l'Université PARIS VI qui nous a donné du temps de calcul pour tester le programme et débuter son exploitation sur l'ordinateur CII 10070 H.

Quel était notre But en élaborant ce programme ?

Tout d'abord automatiser la méthode de composition de la musique stochastique markovienne, c'est-à-dire enchaîner des nuages d'événements sonores (trames) dans le temps.

Ensuite concrétiser l'hypothèse de base (Musiques Formelles p61) que "tout son est une intégration de grains, de particules élémentaires sonores, de quantas sonores". La structure élémentaire des signaux sonores sera donc celle des "grains de GABOR" : fonctions sinusoïdales ayant une enveloppe gaussienne ; leur durée sera fixe.

Le programme est également un outil qui peut permettre certaines recherches sur le "diagramme des courbes d'égale intensité sonore perçue" (Fletcher-Munson). Il est donc une première étape et non un aboutissement.

Il est une étape car il va permettre de "réajuster" certaines hypothèses de travail, d'infirmer ou d'affirmer certains de nos concepts.

Il n'est pas un aboutissement car il est susceptible d'améliorations et d'aménagements considérables dont certains seront donnés en conclusion. Le programme existe déjà actuellement sous deux versions différentes, une troisième est en préparation.

~~Notre~~ seul souci ~~est~~ ici est d'essayer de mettre en lumière les méthodes utilisées dans leur généralité pour parvenir à ces résultats et aussi d'encourager à perfectionner ces méthodes afin d'aller encore plus avant dans la découverte des phénomènes sonores et dans la mise au point de méthodes de composition entièrement nouvelles et encore inexplorées que seul l'emploi de l'outil informatique permet raisonnablement d'envisager.

Je ne voudrais pas terminer cette introduction sans remercier ici toutes les personnes qui m'ont permis de mener à bien ce travail.

En premier lieu, bien sûr, Iannis Xenakis qui m'a continuellement encouragé et fait bénéficier de son expérience.

Bruce Rogers, compositeur, élève de Xénakis, avec qui j'ai débuté le programme et qui est reparti aux USA. ~~Enfin~~

Mlle BESTOUGEFF, Maître de Conférence à PARIS VII pour son assistance technique et le temps machines qu'elle a bien voulu consacrer à ~~mes~~ nos travaux.

PARIS, le 4 Mai 1975.

CHAPITRE I

DESCRIPTION GENERALE

DU PROGRAMME

Le programme ANALOG peut se décomposer, d'une façon très schématique, en quatre parties:

- La première partie consiste à générer la matrice de transition du processus markovien à partir des matrices de transition des fréquences, intensités et densités sonores fournies par l'utilisateur du programme.
- La seconde partie consiste à mettre en œuvre le processus d'une part en itérant ce processus, d'autre part en générant les trames à partir des répartitions données sur les différentes plages de fréquences, intensité et densités sonores
- La troisième partie extrait les quantas un par un dans chaque trame en fonction des caractéristiques propres à chacune d'elle.
- La quatrième partie enfin réalise la liaison entre le programme de composition et la sortie sur bande qui sera exploitée par le convertisseur.

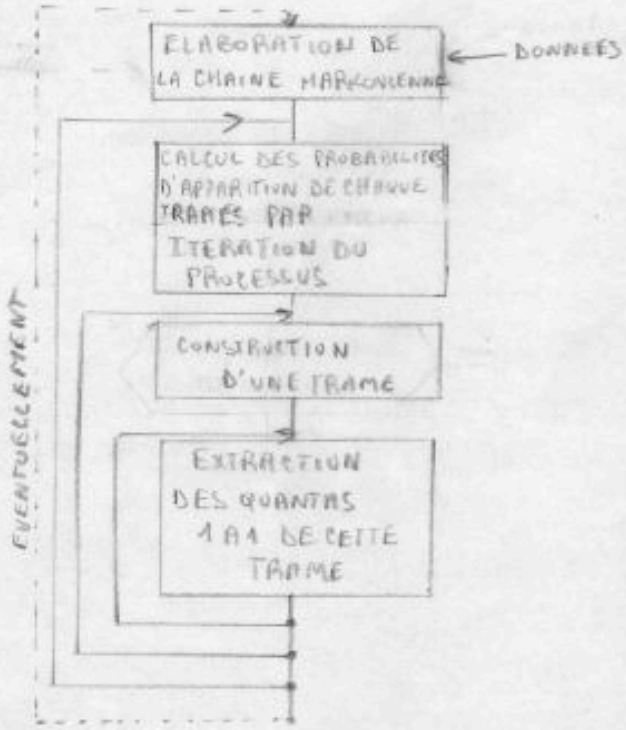


Fig 1

Nous allons examiner successivement chacune de ces parties en partant du niveau macroscopique jusqu'au niveau le plus fin.

1- Génération du processus markovien.

Nous allons ici considérer les trames dans toute leur généralité sans nous soucier de leur contenu, ni même de l'échelle choisie sur chacun des axes. Rappelons simplement qu'en plus d'une "épaisseur" de temps Δt, chaque trame possède est tridimensionnelle.

En effet à l'intérieur d'une trame chaque grain sonore sera positionné par rapport à un axe des fréquences f, un axe des intensités i, et un axe des densités d.

Nous devons donc "jouer" sur ces trois paramètres qui sont les caractéristiques fondamentales de tout événement sonore.

Pour ce faire nous considérerons deux plages de répartition distinctes par paramètres que nous appellerons respectivement : f_1, f_2, i_1, i_2, d_1 et d_2. La façon dont seront réparties les quantités par rapport à ces axes sera décrite plus loin.

Nous définissons donc des matrices de transitions FF1, FF2, I1, I2, D1, D2. qui donnent les probabilités de transition d'une plage à une autre respectivement pour les fréquences, les intensités et les densités

Par exemple pour les fréquences :

FF1 (P)

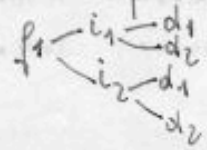
↓	f_1	f_2
f_1	0,3	0,5
f_2	0,7	0,5

FF2 (Q)

↓	f_1	f_2
f_1	0,8	0,3
f_2	0,2	0,7

fig2

Nous sommes donc en mesure de générer le processus car d'une part nous disposons de 8 trames de textures différentes :



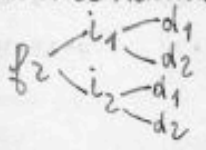


fig3

et d'autre part à l'aide des matrices de transition décrites ci-dessus et d'un couplage adéquat des paramètres nous pouvons calculer toutes les probabilités de transition d'une trame à une autre ce qui nous fournit notre processus markovien qui régule la composition musicale au niveau macroscopique.

Le couplage choisi dans ANALOG est le suivant:

FF1	FF2	D1	D2	I1	I2	I1	I2	FF1	FF2	D1	D2
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
λ	μ	α	β	λ	μ	β	α	γ	ε	γ	ε

avec les correspondances FF1 → α I1 → γ D1 → λ
FF2 → β I2 → ε D2 → μ.

D'autres couplages sont possibles et dans une version ultérieure du programme le choix du couplage sera lui-même déterminé par une loi de probabilité.

A l'aide de ce couplage le calcul de la matrice du processus markovien (MPTZ) devient donc un calcul de probabilités composées et conditionnelles. Cette matrice comportera donc 8 lignes et 8 colonnes et sera telle que pour une colonne données la somme des probabilités soit égale à 1.

Sans entrer dans la théorie des processus markoviens nous pouvons énoncer quelques résultats.

Soit $[\Pi\Pi]$ une matrice de probabilités (ou matrice stochastique) telle que pour une donné $\sum_{j=1}^{n} p_{ij} = 1$ avec $0 \leq p_{ij} \leq 1$ $\forall i,j \in \{1,\dots,n\}$ où p_{ij} est une probabilité indépendante du temps.

Alors si l'on multiplie un vecteur unicolonne v_0 par cette matrice on a: $v_1 = v_0 \cdot [\Pi\Pi]$

et si l'on itère le processus:

$$v_2 = v_1 \cdot [\Pi\Pi]$$
$$v_3 = v_2 \cdot [\Pi\Pi]$$
$$\vdots$$
$$v_k = v_{k-1} \cdot [\Pi\Pi]$$

d'où

$$v_1 . v_2 \ldots \ldots v_k = v_0 . [\Pi] . v_1 . [\Pi] \ldots \ldots v_{k-1} . [\Pi]$$

finalement

$$v_k = v_0 [\Pi]^k$$

La façon dont a été construite la matrice $[\Pi]$ (alias MPIZ) nous permet de dire que nous avons là un processus de Markov ergodique et stationnaire :

- Stationnaire car le processus est indépendant du temps
- ergodique car selon les hypothèses énoncées ci-dessus il existe Π^* telle que :

$$\Pi^* = \lim_{n \to +\infty} [\Pi]^n$$

Cette propriété d'ergodicité est très importante car elle va nous permettre de voir à quel moment l'équilibre est atteint moyennant certains critères.

En effet, au niveau macroscopique, comment le processus va-t-il débuter ?

2. Itération du processus markovien.

Nous allons nous fixer un vecteur unicolonne de départ représentant la répartition des 8 trames définies dans la première partie.

Le processus devant se stabiliser au bout d'un certain nombre d'itérations nous allons choisir de prendre par exemple au départ 100 trames de la forme $(f_1 i_1 d_1)$ soit la trame 1.

Le vecteur unicolonne correspondant est $\begin{pmatrix} 100 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

Ces 100 trames identiques seront jouées au départ.

Ensuite on itère ce vecteur par multiplication avec la matrice MPIZ.

A la première itération les 100 trames de la forme 1 vont se transformer suivant les probabilités de la première colonne de MPTZ.
Un tirage au hasard va permettre d'ordonner l'apparition de chacune des trames en fonction de leur probabilité d'apparition. A nouveau ces trames sont jouées et l'on réitère le processus.

La manière dont est détecté l'état d'équilibre est la suivante.
Soit v_k le vecteur obtenu à la k-ième itération :

$$v_k = (p_1, p_2, p_3, \dots, p_8) \quad \text{avec} \sum_{i=1}^{8} p_i = 100.$$

et soit $v_{k+1} = (p'_1, p'_2, p'_3, \dots, p'_8)$ avec $\sum_{i=1}^{8} p'_i = 100$
le vecteur obtenu à la $(k+1)$-ième itération.

Nous avons choisi, dans le programme, d'arrêter le processus lorsque

$$\text{Max } |p_i - p'_i| < 1 \quad \forall i \in \{1, \dots, 8\}.$$

En effet, l'auditeur ne peut percevoir une différence, dans les trames, aussi minime entre 2 itérations du processus.
Lorsque l'équilibre est atteint, plusieurs choix de poursuite du programme sont possibles. Les utilisations des possibilités seront décrites dans le deuxième chapitre.

Nous allons dire un mot maintenant de la génération des 8 trames servant de base au processus.
C'est en cela que diffèrent les 2 versions actuelles du programme. Seule sera décrite ici la méthode de la version 2 qui fait appel aux probabilités ; la version 1, elle, permettait à l'utilisateur de rentrer directement les répartitions sur les 6 plages de fréquences intensités et densités à l'aide de matrices booléennes ce qui posait un problème lorsque l'on désirait des trames très fines en ce qui concerne les échelles.

Nous entrerons plus en détail sur les données à fournir au programme dans le chapitre II mais schématiquement voici comment sont constituées les trames.

Chacune des 6 plages est divisée en quatre parties égales.
Pour chacune des plages des probabilités sont fournies au programme, ce qui donne quatre probabilités par plage de telle sorte que la somme soit égale à 1. ~~Ensuite le programme répartit sur chacun des segments composant chacune des parties des plages~~ dans

Exemple :

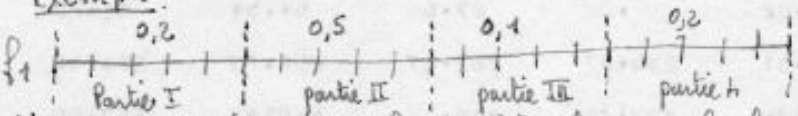


fig4

Nous avons représenté ici la première plage pour les fréquences.
En fonction des probabilités le programme décide d'affecter une fréquen dans la partie II, il choisira l'un des segments de la partie II d'une manière équiprobable. De même pour les autres plages.
Supposons que l'on veuille remplir 5 cases par trame. A la fin du calcul nous aurons par exemple :

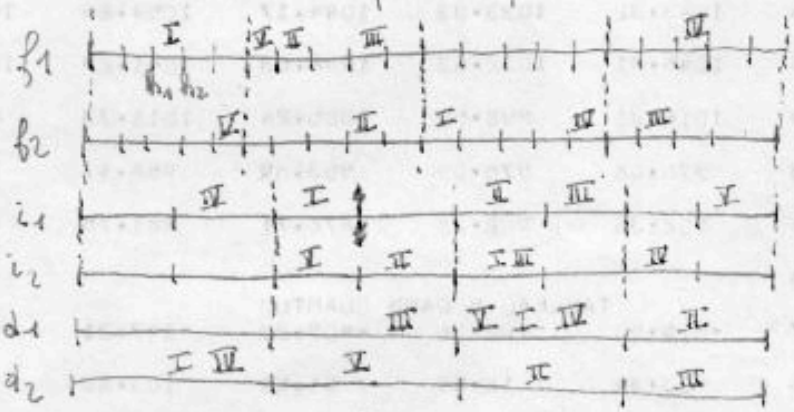


fig5

Il ne reste plus, à partir de ces répartitions sur les différentes plages qu'à reconstituer la texture de chacune des 8 trames à l'aide de la figure 3.

3. Exploitation d'une trame.

A chaque fois qu'une nouvelle trame commence, sa texture est fournie à un sous-programme qui va "passer" les quantas un par un au fur et à mesure de leur calcul à un autre sous programme avant la mise en forme définitive des échantillons sur la bande digitale.

Nous atteignons là le niveau intermédiaire (mésostructure) entre la macrostructure et la microstructure.

Le sous-programme appelé GRILL se compose de deux parties.

La première sert uniquement aux passages des divers paramètres, au calcul de la densité moyenne de la trame et à l'élaboration d'un tableau de probabilités cumulées des densités en fonction des "cases" de la trame qui permet par la suite de définir dans quelle case on doit tirer le quanta à un instant donné.

La deuxième partie est appelée par le sous-programme QUANTA qui "demande" un quanta à chaque fois que le processus le réclame.

La durée de chaque quanta étant fixe, les seuls paramètres à calculer sont : le temps d'attaque, la fréquence et l'amplitude.

Le temps d'attaque est calculé à l'aide de la loi de probabilité élémentaire $f(x).dx = \delta e^{-\delta x} dx$

où δ représente la densité moyenne de la trame considérée.

A ce temps on ajoute le temps d'attaque du précédent quanta de telle sorte que la date d'occurrence d'un quanta considéré reste comprise entre t et $t + \Delta t$ où t représente le temps de départ de la trame considérée et Δt l'épaisseur de temps de cette trame.

Si $T_a > t + \Delta t$ (T_a = temps d'attaque d'un quanta) on considère que la trame est finie et l'on commence la suivante.

La fréquence est calculée à partir du tableau fourni au sous-programme et qui indique la texture de la trame.

En effet reprenons la figure 5 et la plage f_1. Les chiffres romains indiquent les lignes dans lesquelles se trouvent stockées l'information concernant chaque trame.

Par exemple si nous prenons la trame (f_1, i_1, d_1) nous aurons "en machine" le tableau suivant :

(en supposant que les segments de chaque plage soient numérotés)

	f	i	d
I	3	3	3
II	7	5	4
III	9	6	2
IV	18	2	3
V	6	8	3

Voici donc la représentation "en machine" de la trame (f_1, i_1, d_1).

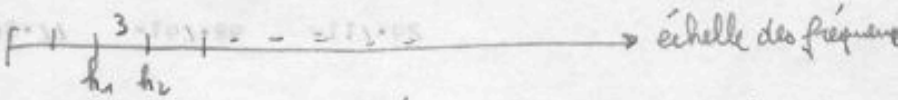
De même pour les autres trames.

A ce stade du programme, la structure est encore très formelle et ne recouvre aucune réalité tangible.

Dans le programme principal sont définies des échelles pour les trois paramètres f, i et d.

Prenons le premier élément de la première ligne: 3.

Cela indique que la fréquence est comprise entre les bornes du 3e segment de l'échelle des fréquences.



Nous assignons donc une fréquence h comprise entre h_1 et h_2 ~~suivant une loi définie au chapitre II, § 7 et elle a une chance équiprobable~~. ~~Si l'échelle est en demi-tons cela n'a aucune importance car nous sommes ici à l'échelon microscopique et seul l'effet de masse nous préoccupe~~.

Nous pratiquons de même avec les intensités.

Ces trois paramètres (temps d'attaque, fréquence, intensité) sont passés au sous-programme QUANTA que nous allons examiner maintenant.

4. Liaison entre le programme de composition musicale et la sortie sur bande

Le sous-programme QUANTA est quasiment le dernier degré de la construction du programme ANALOG. En effet, c'est dans ce sous-programme que sont définis: l'enveloppe et la forme du signal, les pas pour l'échantillonnage et l'addition des différents quantas qui jouent ensemble à un instant donné.

La fonction principale est de découper le temps en "tranches" de la manière suivante:

Représentons un quanta par ⊢ et imaginons que nous ayons une configuration de quanta de la forme:

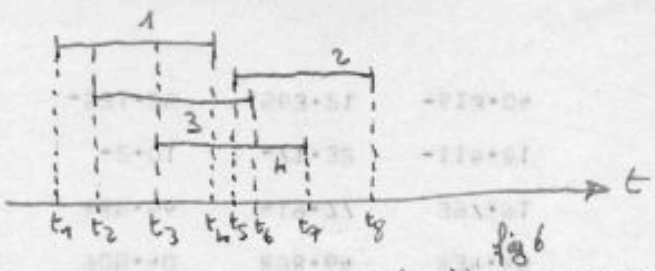


fig 6

Le sous-programme stocke et fournit au sous-programme de sortie sur bande toutes les informations comprises entre 2 évènements (apparition ou disparition d'un quanta.

entre t_1 et t_2 seul le quanta 1 sera joué

entre t_2 et t_3 seront joués 1 et 3 etc---

QUANTA doit calculer les pas pour la fréquence.

Ceci est effectué par l'intermédiaire de la fonction PITCH.

La fréquence origine choisie pour le programme est 16,3508 Hz, ce qui correspond à do_{-1} si $la_3 = 440 Hz$

Le pas est donc défini par la formule: $PAS = 2^{H/12} . XK$

où $XK = (16,3508 \times 512) / SR$.

512 est la dimension du tableau dans lequel est stockée une période de la fonction sinus.

SR est la fréquence d'échantillonnage (13000, 26000, 39000 ou 52000 pts/s pour des mots de 16 bits).

Pourquoi H/12 ? ceci permet tout simplement de "jouer" en demi-tons si H demeure entier.

Le H fourni à PITCH est celui calculé pour les fréquences dans le sous-programme GRILL.

Examinons maintenant la intensité (ou amplitude)

L'intensité (en dB) est en fait un rapport d'intensité logarithmique défini par la formule

$$I_1 - I_2 = 20 \log \frac{W_1}{W_2} \quad (1)$$

Effectuons la correspondance $I2 = 96\,dB \Leftrightarrow W_2 = 2^{15} - 1 = 32767$

En effet 96 dB est l'intensité maximum admise par le convertisseur et $2^{15} - 1$ le nombre maximum machine sur 16 bits

en tenant compte du signe.

Soit I_1 l'intensité d'un quanta, il faut donc chercher la correspondance entre I_1 et w_1 (qui sera la représentation machine de I_1)

de (1) nous tirons

$$I_1 - 96 = 20 \log \frac{w_1}{2^{15}-1}$$

Ceci nous donne $I_1 - 96 = 10(\log w_1 - \log 2^{15}-1)$

d'où $\log w_1 = \log 2^{15} - 1 + \frac{I_1 - 96}{20}$

Passons aux logarithmes népériens (Log) afin de résoudre cette équation

Nous savons que $\text{Log}\, x = \frac{1}{M} \log x$ avec $M = \frac{1}{\text{Log}\, 10} = 0,434294\ldots$

donc

$$M \text{Log}\, w_1 = M \text{Log}\,(2^{15}-1) + \frac{I_1 - 96}{20}.$$

soit $$\text{Log}\, w_1 = \text{Log}\,(2^{15}-1) + \frac{I_1 - 96}{20M}.$$

d'où $$w_1 = 2^{15} - 1\, e^{\frac{I_1 - 96}{20M}}$$

où encore $$\boxed{w_1 = \frac{2^{15}-1}{e^{\frac{96 - I_1}{20M}}}}$$

C'est la formule utilisée dans la fonction DYNAM.

En ce qui concerne l'enveloppe gaussienne pour les quantas les calculs sont en annexe I.

Nous en avons terminé avec la description générale du programme ANALOG ~~est finie~~.

Nous allons examiner plus en détail certains points particuliers ainsi que l'utilisation que l'on peut faire de ce programme.

CHAPITRE II

DESCRIPTION DETAILLEE ET UTILISATION DU PROGRAMME ANALOG.

Nous n'allons pas, dans ce second chapitre, décrire en détail toutes les instructions du programme, ce qui serait long et fastidieux ; nous donnerons seulement les indications nécessaires à la compréhension des données à fournir et certains liens entre les différents sous-programmes utilisés.

Nous ne donnerons aucune explication en ce qui concerne le sous-programme SORTIE qui écrit sur la bande les échantillons, celui-ci est trop spécifique du type de calculateur utilisé.

1- Données concernant les échelles.

a- Echelle des fréquences : (1)

Il faut fournir la fréquence la plus basse (MINF), la fréquence la plus haute ainsi que le degré de finesse sur les fréquences (NFR)

Le format est 3I3. NFR doit être strictement inférieur à 150 et être un multiple de 4 (je rappelle que chacune des échelles est découpée en 4 régions pour des raisons de répartition).

Il faut rappeler également que MINF = 000 $\Leftrightarrow$ fréquence = 16,3508 Hz

$\Leftrightarrow$ Note = do_{-1} (si la_3 = 440 Hz) et qu'une différence égale à 1 équivaut à un ½ ton.

Exemple : si MINF = 001 cela signifie que la fréquence la plus basse utilisée est $do^{\#}_{-1}$ c'est-à-dire 16,3508 Hz $* \sqrt[12]{2}$

De même MAXF dépend de la fréquence d'échantillonnage utilisée. En effet rappelons qu'un théorème attribué à Shannon indique que pour obtenir une fréquence F, la fréquence d'échantillonnage doit être au moins égale à 2F.

Voici à titre indicatif la table des correspondances entre les fréquences différentes vitesses d'échantillonnage du convertisseur et la limite à ne pas dépasser pour MAXF :

SR	MAXF
13000	103
26000	114
39000	121
52000	126

} pour des "mots" de 16 bits.

(1) Ces lettres concernent chaque carte de données et servent à repérer ces cartes pour la clarté de l'exposé.

à l'aide de ces 3 données le programme calcule le pas (PASF pour les fréquences).

PASF = (MAXF - MINF) / NFR

b- Echelle des amplitudes.

Les amplitudes sont directement données en dB et l'on procède de la même façon que pour les fréquences:

Le format pour cette carte est également 3I3.

MINI: ~~minimum d'intensité~~ intensité minimum utilisée en dB (ce minimum peut être égal à zéro).

MAXI: intensité maximum utilisée en dB ~~MAXI ≤ 96~~ (MINI < MAXI ≤ 96)

NIN: degré de finesse de l'échelle des amplitudes (NIN < 50)
NIN doit être un multiple de 4.

De même que précédemment nous avons PASI = (MAXI - MINI) / NIN.

c- Echelle des densités.

Seul est à fournir ici le degré de finesse pour l'échelle des densités (NDE) ce nombre doit être inférieur à 17 et doit être un multiple de 4. Le format pour cette carte est I2.

La formule pour l'échelle des densités est la suivante:

TABDEN(I) = $e^{(I-1)/2}$. Le pas multiplicateur est donc $\sqrt{e}$ pour $I \in \mathbb{N}$ et $I \leq NDE$.

Voir la table en ANNEXE 2.

2- Données concernant certains paramètres particuliers.

a- Il faut fournir DUR, SR, MAXNOT, TLIM, DENTRAM, SIGMA, IALEA.

Le format de cette carte est:

F4.3, F6.1, I4, F5.1, F5.3, F3.2, I8.

- DUR indique la durée des quantas (qui est fixe) en seconde
Exemple: 0040 indique que chaque quanta a une durée de 0,04 s.
- SR indique la fréquence d'échantillonnage.
SR ∈ {13000, 26000, 39000, 52000}.
- MAXNOT indique le nombre de quantas maximum que l'on peut jouer simultanément. On doit avoir MAXNOT ≤ 1000

- TLIM indique la durée maximum en secondes de la partition.
En effet lorsqu'on met en œuvre un processus markovien, on ne sait pas au bout de combien d'itérations il aura atteint l'équilibre. Ceci peut être très long et il faut donc se prémunir contre cette éventua
- DENTRAM indique la densité linéaire des trames.
Exemple : DENTRAM=5 indique qu'il sera joué 5 trames par seco en moyenne, la durée moyenne de chaque trame sera donc de 0,2 seconde dans ce cas précis.
Lors d'un essai simulé en machine 400 trames devaient être jouées, j'avais introduit DENTRAM=5 ce qui donnait une durée de la partition égale à 400/5 = 80 secondes.
La durée effective a été de 77,77295 secondes soit une durée moyenne de 0,1944 s par trame au lieu de 0,2 s.
- SIGMA indique l'écart-type pour l'enveloppe gaussienne des quantu (Voir ANNEXE 1).
- IALEA est un nombre de 8 chiffres, obligatoirement impair, servant à l'initialisation du générateur de nombres aléatoires utilisé dans le programme.

e- Une carte donne le nombre maximum de "cases" remplies par trame (IROM). On doit avoir IROM ≤ 50. Le programme s'arrête par un STOP 111 s'il n'en est pas ainsi.

3- Données concernant les répartitions sur les six plages f1, f2, i1, i2, d1, d2.
Nous appellerons les 6 cartes f1, f2, f3, f4, f5, f6 avec la correspondance suivante :

f1 ⟷ f1
f2 ⟷ f2
f3 ⟷ i1
f4 ⟷ i2
f5 ⟷ d1
f6 ⟷ d2

Le format pour chaque carte est 4F4.3.
Ce sont, pour chacune des plages les probabilités figurant mises en évidenc sur la figure 4 et qui permettent la répartition des quantas

dans les différentes trames. Nous verrons plus loin comment est effectuée la liaison entre ces répartitions et la représentation "machine" des trames.

4- Données concernant le processus markovien.

Il s'agit là encore de 6 cartes que nous appellerons g1, g2, g3, g4, g5, g6 dont le format est 4F4.3 et qui donnent les 6 matrices de transition de la forme de celles décrites sur la figure 2.

Nous avons la correspondance

cartes	Tableaux matriciels
g1	⟷ FF1
g2	⟷ FF2
g3	⟷ I1
g4	⟷ I2
g5	⟷ D1
g6	⟷ D2

Les matrices sont lues sur les cartes ligne par ligne.

A partir de ces matrices et du couplage (actuellement programmé), le programme génère la matrice du processus markovien (MPTZ).

5. Choix de poursuite du programme.

Lorsque le programme détecte l'état d'équilibre, il faut lui indiquer ce qu'il doit faire ensuite à l'aide de la carte h.

Cette carte a pour format I1 et sert à alimenter l'indicateur M.

Quatre choix sont possibles:

M=1 ⇒ on lit de nouveau 6 cartes de la forme f (ou g3 répartitions sur les plages). On change donc la texture des trames.

M=2 ⇒ on lit de nouveau 6 cartes de la forme g (g4, matrices de transition), on ne change pas la texture des trames, mais simplement le processus markovien (MPTZ).

M=3 ⇒ sans changer les données, on calcule une nouvelle permutation; si la première était V=(100,0,0,0,0,0,0,0) la seconde sera V=(0,100,0,0,0,0,0,0) c'est-à-dire 100 trames de la forme (f1, i1, d2).

M=4 ⇒ le programme s'arrête par un STOP 200.

Remarque: le programme s'arrête par un STOP 100 si TLIM est dépassé.

En tout état de cause la dernière carte de donnée doit toujours être du type h.

Il ne m'a pas semblé souhaitable d'introduire une cinquième possibilité qui aurait consisté à changer à la fois la texture des trames et le processus markovien. En effet dans ce cas, l'unité de l'œuvre serait détruite tous les paramètres étant changés au niveau de la macrocomposition.

6- Description détaillée de la génération des trames à partir des données.
Voir les parties du programme correspondante en ANNEXE 3.

a) SUBROUTINE DONTRM

6: la boucle 1500 est effectuée pour chacune des 6 plages.

7: lecture d'une carte de la forme f

9-10: cumul des probabilités lues.

11: la boucle 1000 est effectuée autant de fois qu'il y a de "cases" à remplir dans les trames.

12: NCASE grâce à la fonction IVRN (principe du tirage de boules de couleurs différentes dans une urne avec remise) indique la région (de 1 à 4) dans laquelle se trouve la "case" considérée.

13: LLI = 1 ou 2 : répartition pour les fréquences.
LLI = 3 ou 4 : " " " amplitudes
LLI = 5 ou 6 : " " " densités.

14 à 21: en fonction du degré de finesse pour chacun des paramètres (NFR, NIN, NDE) on calcule le nombre de segments par région (JFF) et l'équiprobabilité correspondante (PROBA)

22-23: PEQUI: tableau cumulé des équiprobabilités en fonction de JFF et PROBA.

24: toujours à l'aide de la fonction IVRN on calcule l'emplacement (IT) de la "case" par rapport au segment choisi (NCASE).

25: Aiguillage suivant la plage que l'on est en train de traiter.

26-28-30-32-34: On remplit les différents tableaux suivant la plage traitée permettant de garder les répartitions des "cases" remplies sur les différentes plages.

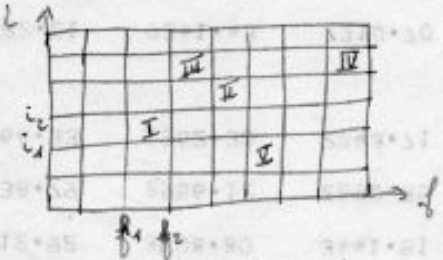
Ce sous-programme étant exécuté, les tableaux LF1, LF2, LI1, LI2, LD1, LD2 donnant la répartition sur chaque plage, sont

donc alimentés.

Il ne reste plus qu'à constituer les trames, ce qui est fait en appelant le sous-programme TXTRAM (cartes 60 à 67 dans le programme principal).

7. Calcul d'une fréquence et de l'amplitude pour un quanta déterminé.

Soit la trame :



Supposons que l'on sait au stade du calcul des paramètres d'un quanta, c'est-à-dire que l'on sache dans quelle région de la grille trame il doit se trouver (par exemple la case I).

Pour le calcul de la fréquence (le problème est identique pour l'amplitude) le problème consiste à trouver la probabilité pour qu'un segment s, intérieur à un segment de droite de longueur a, ait une longueur comprise entre h et h+dh ($0 \le h \le a$)

Cette probabilité est donnée par la formule :

~~$P_h = \theta(h)\,dh = \frac{2}{a}\left(1-\frac{h}{a}\right).dh$ (1)~~

$$P_h = \theta(h)\,dh = \frac{2}{a}\left(1-\frac{h}{a}\right).dh \quad (1)$$

avec ici $a = f_2 - f_1$.

Le calculateur ne pouvant tirer que des nombres y au hasard suivant la loi d'équiprobabilité ($0 \le y \le 1$), pour un intervalle quelconque h_0 nous avons :

$$\text{prob}(0 \le y \le h_0) = \int_0^{h_0} \theta(h)\,dh = \frac{2h_0}{a} - \frac{h_0^2}{a^2} = F(h_0)$$

Mais $F(h_0) = \text{prob.}(0 \le y \le y_0) = y_0$

- d'où $\frac{2h_0}{a} - \frac{h_0^2}{a^2} = y_0$ donc $h_0 = a\left(1 \pm \sqrt{1-y_0}\right)$

Rejetons la racine munie du signe + car h_0 doit être inférieur à a, on obtient donc :

$$h = a\left(1-\sqrt{1-y}\right) \quad \forall y : 0 \le y \le 1$$

(1) Musiques formelles p43.

Une fonction dans le programme réalise donc ce calcul en conservant, en plus la mémoire de la fréquence du quanta précédent qui a été tiré dans la même case.

Soit h_{n-1} la fréquence du quanta précédent dans la case I, la fréquence h_n que l'on calcule est donc :

$$h_n = h_{n-1} \pm (f_2 - f_1)(1 - \sqrt{1-y}) \quad \text{avec } 0 \le y \le 1.$$

La fonction + ou - est également une fonction du programme appelée COIN (principe du pile ou face).

Le même type de calcul est réalisé pour les amplitudes.

Remarque : il se peut que de cette façon nous n'ayons pas toujours $f_1 \le h_n \le f_2$, nous décidons d'effectuer ce calcul un certain nombre de fois (5 au maximum) pour "tomber" dans l'intervalle $[f_1, f_2]$, si cela n'est toujours pas réalisé, alors on prend $h_n = f_1 + y(f_2 - f_1)$ avec $0 \le y \le 1$.
(fonction uniforme).

8- Remarque concernant les échantillons écrits sur bande.

Il se peut qu'à un instant donné, le nombre à inscrire sur la bande soit supérieur, en valeur absolue, à 32767 ($2^{15}-1$), par suite de l'addition de nombreux quanta. Dans ce cas on prend l'image miroir par rapport, selon le cas, aux droites $y = \pm 32767$.
Ceci se produit assez rarement du fait de la loi utilisée pour les dates d'occurrence de chaque quanta, même si la densité est élevée.

9- Remarques générales concernant le programme ANALOG.

Outre le programme principal, il comporte 11 sous-programmes, dont certains possèdent plusieurs entrées et 10 fonctions.

Ces sous-programmes sont :

TXTRAM : constitution des 8 trames à partir des répartitions sur les six plages f_1, f_2, i_{f_1}, i_{f_2}, d_1, d_2. (voir ANNEXE 3)
DONNEE : lecture des cartes de type a, b etc et détermination des pas sur l'échelle des fréquences et celles des amplitudes.
DONTRM : lecture des probabilités de répartition sur les plages et constitution de ces répartitions (voir ANNEXE 3)
XITER : Multiplication du vecteur unicolonne par la matrice des processus markovien (MPT2).

PERT : calcule une nouvelle perturbation, c'est-à-dire un vecteur dont toutes les composantes sont à zéro sauf une.

GRILL1 : plusieurs entrées, calcule les paramètres propres à chaque quanta pour une trame donnée, calcule aussi la densité moyenne de celle chaque trame.

QUANTA : découpe le temps en "tranches", calcule le nombre d'échantillons à sortir sur la bande pour chaque Δt, calcule également différents pas. Possède plusieurs entrées.

QUANTUM : balaie l'enveloppe gaussienne et la sinusoïde en fonction des pas calculés dans QUANTA et additionne les quanta.

GAUS : calcule, en fonction de SIGMA, l'enveloppe gaussienne et la stocke dans un tableau à 512 positions.

SINA : stocke dans un tableau à 512 positions une période de la courbe $y = \sin x$.

SORTIE : Plusieurs entrées.
- initialise le buffer au départ pour le convertisseur
- transforme les mots de 36 bits en mots de 16 bits acceptables par le convertisseur.
- Prend l'image suivante s'il y a lieu.

Les fonctions sont :

TUNE : effectue le calcul décrit au §7.

IURN : ~~Image de la~~ principe du tirage des boules de couleurs dans une urne avec remise. Sert principalement au tirage des trames une à une à chaque itération du processus markovien

XLINER : sous-fonction de la fonction TUNE

UNIFRM : fonction de probabilité uniforme.

EXPON : utilisée pour calculer les dates d'occurence des quanta. Utilise la loi de probabilité $f(x).dx = \delta e^{-\delta x} dx$

COIN : principe du pile ou face - Donne les valeurs -1 ou +1.

RANF : générateur de nombres aléatoires compris entre 0 et 1.

ORIGIN : calcule à partir de quelle abscisse x ($0 \leq x \leq 512$) on doit balayer la sinusoïde de telle manière que le sommet de celle-ci, pour une fréquence donnée, coïncide avec le sommet de l'enveloppe gaussienne. Ceci afin d'avoir l'énergie sonore maximum pour chaque quanta.

PITCH : détermine le pas de balayage de la sinusoïde en fonction de la fréquence.

DYNAM : convertit les dB en nombre machine (voir fin du chap. I).

Précisons que le programme requiert une allocation mémoire d'environ 120K pour son exécution.

Le temps CPU dépend beaucoup de la densité moyenne de quantas lus d'un passage. Pour 700 q/s il faut environ 1 heure de CPU pour 1 mn de musique sur le CII 10070.

Le rapport est grossièrement divisé par 6 sur un IBM 370/168.

CHAPITRE III

PREMIERES CONCLUSIONS
PRINCIPAUX AXES DE RECHERCHE
DEVELOPPEMENTS ULTERIEURS
DU PROGRAMME.

1- Développements ultérieurs du programme ANALOG.

a- au niveau de la microcomposition.

Une troisième version du programme va permettre d'obtenir des grandeurs de durée variable. Cela ne constitue pas un progrès essentiel, à mon avis, si l'on exploite le programme avec des densités relativement élevées (au minimum 150 à 200 g/s). Par contre cette question devient essentielle si l'on utilise des densités faibles (événements rares) du type 1 à 10 grains/s.

Le problème consiste en la construction d'un algorithme permettant, pour chaque grain, de calculer sa durée.

Si nous restons dans le domaine microcompositionnel nous pouvons utiliser une loi de répartition probabiliste du type de celle utilisée pour calculer les temps d'attaque des grains. Je ne pense pas que cela soit l'esprit du programme ANALOG.

L'algorithme en question doit se placer dans le cadre macrocomposition et le paramètre de durée doit être traité au même titre que les trois autres : fréquence, amplitude, densité.

Alors on peut également définir deux plages de répartition pour la durée t_1 et t_2 et introduire deux nouvelles matrices de transition paramétrées. Dans ce cas les macro-images sonores (les trames) seront au nombre de 16 et MPTZ aura la dimension pour dimensions 16x16.

De toute façon, l'introduction de la variabilité de la durée des grains modifie l'objectif premier du programme qui est de montrer que "tout son est une intégration de particules élémentaires sonores".

b- au niveau de la macrocomposition.

Actuellement le couplage entre les différentes matrices de transition FF1, FF2, I1, I2, D1 et D2 est fixe et "programmé". Or il existe de nombreuses façons de coupler les matrices. Il pourrait donc être envisagé de paramétrer ce couplage à l'aide de certains critères, ou tout simplement de matrices de probabilités. Ceci ne paraît pas essentiel dans

l'état actuel de mes recherches.

Par contre la création des répartitions sur les six plages f_1, f_2, i_1, i_2, d_1 et d_2 pourrait être envisagée sous un angle très différent. En effet, parallèlement au processus markovien qui régit l'enchaînement des trames dans le temps d'une façon stochastique, il serait certainement intéressant et enrichissant de coupler un processus déterministe pour constituer ces plages et par là même la texture des trames. Je pense ici à la théorie des cribles et aux métaboles qui permettent de les enchaîner dans le temps [1]. Les deux procédés compositionnels, aléatoire et déterministe, si longtemps ennemis trouveraient là tout naturellement un terrain pour se confronter et pourquoi pas s'allier et se compléter pour le plus grand bénéfice de la composition musicale automatique et de la pensée contemporaine en général.

A un niveau encore plus général, il faudrait étudier l'incidence des entropies de chacune des matrices de transition sur le processus markovien (MPTZ) et donc sur l'enchaînement des trames. Rappelons ici, comment mesurer l'entropie d'une matrice de probabilités [2]

La définition de l'entropie d'un système est:

$$H = -\sum p_i \log_2 p_i \quad \text{si } H \text{ est mesurée en bits.}$$

Le calcul de l'entropie d'une matrice du type FF2 par exemple (fig2 chap) se fait d'abord par colonnes ($\sum p_i = 1$), p_i représentant un élément de FF2.

Reprenons cette matrice:

↓ \	f_1	f_2
f_1	0,8	0,3
f_2	0,2	0,7

Nous aurons:

entropie des états de f_1:

$$H_{f_1} = -0,8 \log_2 0,8 - 0,2 \log_2 0,2 = 0,7219 \text{ bits}$$

entropie des états de f_2

$$H_{f_2} = -0,3 \log_2 0,3 - 0,7 \log_2 0,7 = 0,8813 \text{ bits}$$

(1) cf [illegible] Iannis Xenakis in "LA NEF" n°29: "Vers une métamusique".
(2) Xenakis in "Musiques Formelles" p 104 et 80 à 82.

Nous devons maintenant calculer les valeurs, à l'équilibre, de la des valeurs états f_1 et f_2.

A l'équilibre nous devons avoir :

$$\begin{pmatrix} 0,8 & 0,3 \\ 0,2 & 0,7 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \quad \text{avec} \quad f_1 + f_2 = 1 \ (3)$$

d'où $0,8 f_1 + 0,3 f_2 = f_1$ (1)

et $0,2 f_1 + 0,7 f_2 = f_2$ (2)

ona :

$-0,2 f_1 + 0,3 f_2 = 0$ (1)

$0,2 f_1 - 0,3 f_2 = 0$ (2)

Remplaçons l'une des 2 équations par (3)

$f_1 + f_2 = 1$

$0,2 f_1 - 0,3 f_2 = 0$

d'où nous tirons les valeurs à l'équilibre :

$f_1 = 0,6$ et $f_2 = 0,4$

L'entropie moyenne à l'équilibre est :

$H_m = 0,7219 \times 0,6 + 0,8813 \times 0,4 \simeq 0,785$ Bits.

Voir ANNEXE 4 pour les entropies de différentes matrices.

Pb1 au lieu de choisir des paires de matrices de transition nous pourrions alors choisir des paires d'entropies ce qui permettrait grâce à un tableau du type de celui de l'annexe 4 de choisir correctement les matrices en fonction du degré d'ordre ou de désordre voulu. J'ai réalisé partiellement cette correspondance, il faudrait maintenant descendre à un niveau plus fin et calculer également l'entropie de la MPT 2 obtenu

Pb2 Nous entrons ici dans un domaine plus général que le cadre du présent document, mais il serait très intéressant de savoir si cette mesure par les entropies correspond réellement à un critère "esthétique" dans le domaine de la composition musicale stochastique.

2. Premières conclusions.

A l'écoute des premiers essais réalisés à l'aide de ce programme il apparaît que cette méthode qui consiste à générer un univers sonore à partir de grains élémentaires, s'avère très prometteuse pour deux raisons.

La première étant qu'elle ne fait pas appel à l'analyse de Fourier et

qu'elle semble plus générale Bien que moins "maîtrisable" actuellement. Pour la maîtriser mieux il faudrait explorer d'une manière assez systématique les différentes textures des trames (en particulier en se référant au diagramme des courbes d'égales intensités de Fletcher-Munson et en approximant ces courbes ; Voir ANNEXES). Cela prendra du temps mais deviendra nécessaire très vite si l'on veut progresser.

La deuxième raison est le fait que, même sans cette recherche, elle permet d'obtenir des résultats sonores complexes sans grand effort, ce qui n'est pas le cas avec l'analyse de Fourier. Pour résumer on pourrait dire que cette méthode permet de "s'évader" plus facilement des schémas classiques (harmoniques et autres) que l'analyse de Fourier ne le permet.

La première partie du programme peut (création de la MPTZ à partir des matrices de transition) peut être isolée afin de générer de nombreux processus markoviens et de les analyser, en particulier calculer le nombre d'itérations nécessaires pour atteindre l'équilibre (voir chap I) afin d'utiliser telle ou telle matrice dans les compositions ultérieures suivant que l'on veut obtenir des états d'équilibre très rapides ou au contraire des évolutions plus lentes.

Vous voyons là, l'apport incontestablement bénéfique de l'outil informatique dans l'analyse, la création et la recherche de formes musicales nouvelles et complexes que seul la machine est capable d'engendrer. Rappelons que la pièce de I. Xenakis ANALOGIQUE basée sur les processus markoviens mais faite "à la main" fut créée en 1959 à Gravesano et qu'elle durait 2'30". Il aura fallu 16 ans avant que cette méthode de composition ne soit automatisée et que l'on commence ainsi à en explorer toutes les richesses.

3- Principaux axes de recherche.

Entre les développements possibles du programme ANALOG ici décrit, plusieurs voies certainement fructueuses sont ouvertes.
Je citerai d'abord celles dans lesquelles je me m'engagerai pas et cette liste ne prétend pas être exhaustive.

- Recherche de timbres à l'aide de MUSICS ou d'autres outils créés ou à créer.
- Analyse d'œuvres passées (musicologie, ethnomusicologie, etc...)
- Création en temps réel d'échantillons sonores avec interaction entre le "compositeur" et la machine.
- Réalisation de partitions graphiques "traditionnelles". Ceci est d'ailleurs réalisé maintenant (9).
- Analyse spectrale des sons.

(9) Cf Don Byrd in "Random Bits" Mass Wrubel Computing Center, Indiana University Bloomington, Mars 75.

Par contre tout ce qui a trait à la composition musicale elle-même retient toute mon attention et en particulier le domaine de la macro-composition. A l'évidence, les ces différentes branches ne peuvent être complètement dissociées et les découvertes de de l'une ou l'autre de celles-ci rejaillissent sur celles-là. Il n'en demeure pas moins vrai que une méthode compositionnelle est indépendante des "outils" utilisés et qu'elle conserve toute sa généralité quelle que soit l'évolution technologique à venir. L'exemple de ANALOGIQUE A et le montre bien, ANALOGIQUE A ayant été créée pour 9 cordes, ANALOGIQUE B étant une œuvre électro-acoustique et maintenant ce procédé compositionnel est adapté pour un convertisseur Digital / Analogique. De plus elle une méthode de composition musicale peut être adapté à d'autres domaines de l'art et en particulier au domaine lumino-cinétique (cf le Polytope), ou à la peinture ou au dessin voire à la sculpture.

En préambule aux différents axes possibles concernant la composition musicale automatique, rappelons ici le point de départ théorique d'une utilisation des calculatrices électroniques en composition musicale tel que le précise I. Xenakis (1):

a- La pensée créatrice de l'homme secrète des mécanismes mentaux qui ne sont, en dernière analyse, que des ensembles de contraintes, de choix, et ceci dans tous les domaines y compris les arts;

b- Certains de ces mécanismes sont mathématisables;

c- Certains de ces mécanismes sont réalisables physiquement (sons, moteurs, fusées, machines à calculer digitales, analogiques, etc...);

d- Certains mécanismes mentaux peuvent trouver des correspondances avec certains mécanismes de la nature;

e- Certains aspects mécanisables de la création artistique peuvent être simulés par certains mécanismes physiques (machines) existants ou à créer;

f- Il se trouve que les ordinateurs peuvent rendre certains services.

Je me permettrai ici de commenter et souligner quelques points.

Remarquons tout d'abord le ton volontairement prudent de ces assertions, sans cesse soumises à une impossibilité hypothétique (certains). En effet, l'acte de création [illegible] ~~entre~~

(1) Musiques Formelles p 165.
(2) Art et science: "De la ~~part et créativité~~.

processus est issu d'un processus mental excessivement complexe et "tout" n'est pas possible immédiatement. Les contraintes et les choix sont multiples et essentiellement de deux ordres: ceux librement consentis (choix de départ entre plusieurs théories ou à l'intérieur d'une théorie, adaptabilité de ceux-ci à l'idée directrice ou schèmes), ceux qui sont imposés consciemment ou inconsciemment :

Consciemment: connaissance insuffisante de tel ou tel domaine, expérience.

Inconsciemment: poids du passé, traditions, ~~contraintes~~, inhibitions etc...

~~Les raisons qui m'ont poussé à faire ce premier travail. Des nécessités~~

Après avoir effectué ces choix il reste à faire le point. Certains sont mathématisables et même pourrions-nous dire mathématisés, d'autres ~~sont~~ sont réalisables physiquement, pour d'autres encore certaines correspondances "naturelles" peuvent être mises en évidence. Il résulte donc de ceci que certains aspects de la création artistique peuvent être simulés à l'aide de machines. Voyons donc maintenant les choix et les contraintes ~~que je me suis~~ qui ~~fixés~~ se posent pour mes recherches ultérieures :

~~Tout d'abord les choix~~:

- Produire des événements sonores uniquement à l'aide des conversions Digitale/analogiques.
- Dans une première phase, n'utiliser que les quantas tels qu'ils sont définis dans les chapitres précédents ou quelque peu modifiés.
- Rechercher tout algorithme permettant de créer une structure macrocompositionnelle. Ces algorithmes pouvant être déterministes ou aléatoires ou les deux.
- Faire en sorte que chacune des méthodes macrocompositionnelles ainsi définies soient immédiatement distinguables du point de vue auditif les unes des autres.

Ces choix entraînent une contrainte technologique : celle de disposer, en général de gros calculateurs car la masse d'informations à manipuler, canaliser, maîtriser est considérable. En effet, pour avoir, ne serait-ce qu'au niveau de la fréquence, une définition sonore recouvrant tout le spectre audible il faut pouvoir échantillonner au moins jusqu'à 35 000 points par secondes. Cela nécessite également un temps considérable de réflexion pour arriver à une parfaite adéquation entre la théorie, le plus souvent mathématique, et l'esthétique, ou du moins, le résultat sonore et les buts à atteindre

Il est hors de question ici d'entrer dans le détail des différents algorithmes permettant de créer des structures micro-compositionnelles. Nous pouvons examiner cependant les lignes directrices de quelques-uns d'entre eux.

Tout d'abord certaines idées émises par Xénakis dans "Musiques Formelles" ou la revue d'esthétique attendent encore d'être informatisées; il s'agit notamment de la musique symbolique à l'aide de la théorie des ensembles (Herma pour piano) et de la théorie des cribles modulo 3. (Nomos Alpha, Akrata) mais aussi de la stratégie musicale à l'aide de la théorie des jeux (Duel, Stratégie) de Xénakis, Tenso for Trombone and Tuba de B. Rogers).

Outre ces domaines déjà plus ou moins formalisés il reste de nombreux champs d'investigation liés à la théorie des graphes (dont les chaînes de Markov sont un domaine annexe) et à la théorie des automates et aux grammaires transformationnelles. Je me propose d'aborder ce domaine de la composition musicale à l'aide de la théorie des automates et des grammaires.